UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/766,336	01/28/2004	Shyh-Mei F. Ho	SVL920030102US1	3271

45112 7590 01/10/2008
Kunzler & McKenzie
8 EAST BROADWAY
SUITE 600
SALT LAKE CITY, UT 84111

EXAMINER	
ALVESTEFFER, STEPHEN D	
ART UNIT	**PAPER NUMBER**
2173	
MAIL DATE	**DELIVERY MODE**
01/10/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

AK

Office Action Summary	**Application No.** 10/766,336	**Applicant(s)** HO ET AL.	
	Examiner Stephen Alvesteffer	**Art Unit** 2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 October 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-20,22-29 and 31-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-20,22-29 and 31-40* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

This Office Action is responsive to the amendment filed October 26, 2007. Claims 1, 4, 6, 7, 20, 22, 24, 25, 29, 39, and 40 are amended. Claims 3, 21, and 30 are cancelled. Claims 1, 12, 20, 29, 39, and 40 are independent. Claims 1, 2, 4-20, 22-29, and 31-40 remain pending.

Claim Objections

Claims 7 and 25 objected to because of the following informalities: "command-line interface" lacks antecedent basis. "command-line interface" should be corrected to –interface—. Appropriate correction is required.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 4-6, and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bach et al. (hereinafter Bach '739), United States Patent number 5,781,739, and Bach et al. (hereinafter Bach '660), United States Patent number 6,141,660.

Regarding claim 1, Bach '739 teaches an apparatus for automatically generating a web interface for an MFS-based IMS application (see column 1 lines 55-

57; "*...translate MFS source for the purpose of running an IMS transaction*"), comprising: an import module configured to import MFS-based IMS source files corresponding to an MFS-based IMS application (see column 2 lines 62-66; "*...browse and download MFS source information*"); a metadata generator configured to store a standardized metadata description of the Message Input Description (MID) and Message Output Description (MOD) for the MFS-based IMS application (see column 4 lines 51-59; "*IMS Web uses only the information in MFS Message Input Descriptors (MIDs-111) and MFS Message Output Descriptors (MODs-112) to format input and output messages*"); and a code generator configured to generate a middleware application corresponding to the MFS-based IMS application from the standardized metadata description, the middleware application interfacing between a client application and the corresponding MFS-based IMS application (see column 10 lines 44-49; "*The generated CGI-BIN program invokes this class to parse the input string from the Web browser*", code is generated to interface between a web browser and the IMS database). Bach '739 does not teach an interface configured to execute the import module, the metadata generator, and the code generator in response to a parameter set provided as a single input to the interface. Configuring an interface to execute several different instruction sets in response to a parameter set provided as a single input was well known in the art at the time the invention was made. Bach '660 teaches generating HTML input and output forms from IMS database descriptions using a command-line interface (see Bach '660 column 17 lines 4-12; "*To begin using the command-line interface, the user goes to the directory in which the product was installed, and enters* *the command "IOCCLI." A command prompt, such as "COMMAND >>", is then displayed on the monitor and the user can begin entering CDT 400 commands*"). Furthermore, Bach teaches using batch processing to execute several functions at once in response to a parameter set provided as a single input to the interface (see Bach '660 column 17 lines 54-67; "*The RUNSCRIPT command runs a file that contains all of the necessary commands. A command script can be created either by using the CDT 400 GUI 402 or by saving the script when prompted while using the QUIT command. The option of allowing for batch processing with the RUNSCRIPT command means the user no longer has to go through all the panels of the CDT 400 GUI 402 and can simply modify script files (or write their own) and run them through the CLI 403. In theory, this should save them a lot of time (the user may not want to go through all the GUI 402 panels for each database if they have a substantial number of databases). Users can also concatenate several script files into one and just run that single file through the CLI.*"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to perform the operations as described by Bach '739 on a command line interface using batch files as described by Bach '660 for the purpose of automating tedious multiple-command entry on an interface.

Regarding claim 4, Bach '739/'660 teaches a loader configured to load a script comprising the parameter set from persistent storage (see Bach '660 column 17 lines 4-12; "*the user can enter commands one by one, or can run a command script that contains all of the commands*").

Regarding claim 5, Bach ‘739/’660 teaches that the script comprises a plurality of parameter sets each associated with a different MFS-based IMS application (see Bach ‘660 column 17 lines 4-12; “*the user can enter commands one by one, or can run a command script that contains all of the commands*”, the command line interface described by Bach ‘660 is capable of comprising a plurality of parameter sets each associated with a different MFS-based IMS application).

Regarding claim 6, Bach ‘739/’660 teaches that the parameter set is manually entered, the apparatus further comprising a storage module configured to store the manually entered parameter set for subsequent automated use (see Bach ‘660 column 17 lines 54-67; “*A command script can be created either by using the CDT 400 GUI 402 or by saving the script when prompted while using the QUIT command. The option of allowing for batch processing with the RUNSCRIPT command means the user no longer has to go through all the panels of the CDT 400 GUI 402 and can simply modify script files (or write their own) and run them through the CLI 403*”).

Regarding claim 11, Bach ‘739/’660 teaches a deployment module configured to store the standardized metadata descriptions and middleware application in one or more repositories (see Bach ‘739 paragraph [0014]; “*Places the CGI-BIN executable file in the directory which was user specified in IMS Web Studio (`Web Server's CGI Path`). This directory must be the same directory that their Web server uses to look for CGI-BIN scripts*”, the CGI-BIN directory is the repository that stores the generated middleware applications to interface with the MFS-based IMS applications).

Claims 2, 12-15, 19, 20, 22-24, 29, 31-33, and 38-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bach '739 (5,781,739) *supra*, Bach '660 (6,141,660) *supra*, and Francis et al. (hereinafter Francis), United States Patent number 6,665,861.

Regarding claim 2, Bach '739/'660 teaches every limitation of claim 2 except that the standardized metadata description comprises an eXtended Markup Language Metadata Interchange (XMI) file. Francis teaches exchanging data between web applications and database applications using XMI (see Francis column 6 line 59 through column 7 line 22; "*these models are persisted in an XML Metadata Interchange (XMI) form although other forms may be used without departing from the spirit and scope of the present invention. XMI is a new standard from the Object Management Group (OMG) that combines the Unified Modeling Language (UML) with the Extensible Markup Language (XML) standard. This new open industry standard combines the benefits of the web-based XML standard for defining, validating, and sharing document formats on the web with the benefits of the object-oriented Unified Modeling Language (UML), a specification of the OMG that provides application developers a common language for specifying, visualizing, constructing, and documenting distributed objects and business models*"). XMI was a well-known standard for exchanging data at the time the invention of the instant application was made. It would have been obvious to one of ordinary skill in the art at the time the invention was made to exchange the data in the application taught by Bach '739/'660 using XMI as taught by Francis in order to exchange data while taking advantage of the benefits of both XML and UML.

Claim 12 recites an apparatus with substantially the same limitations as the apparatus of claim 2. Therefore, claim 12 is rejected under the same rationale.

Regarding claim 13, Bach '739/'660/Francis teaches a loader configured to load a script comprising the parameter set from persistent storage (see Bach '660 column 17 lines 4-12; "*the user can enter commands one by one, or can run a command script that contains all of the commands*").

Regarding claim 14, Bach '739/'660/Francis teaches that the parameter set is provided as a single input to the command-line interface (see Bach '660 column 17 lines 4-12; "*To begin using the command-line interface, the user goes to the directory in which the product was installed, and enters the command "IOCCLI." A command prompt, such as "COMMAND >>", is then displayed on the monitor and the user can begin entering CDT 400 commands*").

Regarding claim 15, Bach '739/'660/Francis teaches that the middleware application comprises a server component and a back-end component (see Bach '739 Figure 2 and column 4 lines 17-27; "*The source files are then used to develop a corresponding input HTML form and to create the classes necessary to build the CGI-BIN program 105 located on a Web server 106. Once an input HTML form 102 is created it may be populated by a user to request specific information which may be located within one or more databases located within the IMS*", the CGI-BIN program is equivalent to middleware. The Web server is the server component. The databases located within the IMS is the back-end component.).

Regarding claim 19, Bach ‘739/‘660/Francis teach a deployment module configured to store the XMI files and middleware application in one or more repositories (see Bach ‘739 paragraph [0014]; “*Places the CGI-BIN executable file in the directory which was user specified in IMS Web Studio (`Web Server's CGI Path`). This directory must be the same directory that their Web server uses to look for CGI-BIN scripts*”, the CGI-BIN directory acts as a repository to store the generated middleware applications that interface with the MFS-based IMS applications).

Claim 20 recites a utility with substantially the same limitations as the apparatus of claim 14. Therefore, claim 20 is rejected under the same rationale.

Regarding claim 22, Bach ‘739/’660/Francis teaches a loader configured to load a script comprising the parameter set from persistent storage (see Bach ’660 column 17 lines 4-12; “*the user can enter commands one by one, or can run a command script that contains all of the commands*”).

Regarding claim 23, Bach ‘739/’660/Francis teaches that the script comprises a plurality of parameter sets each associated with a different MFS-based IMS application (see Bach ‘660 column 17 lines 4-12; “*the user can enter commands one by one, or can run a command script that contains all of the commands*”, the command line interface described by Bach ‘660 is capable of comprising a plurality of parameter sets each associated with a different MFS-based IMS application).

Regarding claim 24, Bach ‘739/’660/Francis teaches that the parameter set is manually entered, the apparatus further comprising a storage module configured to store the manually entered parameter set (see Bach ‘660 column 17 lines 54-67; “*A* *command script can be created either by using the CDT 400 GUI 402 or by saving the script when prompted while using the QUIT command. The option of allowing for batch processing with the RUNSCRIPT command means the user no longer has to go through all the panels of the CDT 400 GUI 402 and can simply modify script files (or write their own) and run them through the CLI 403*").

Claim 29 recites a method with substantially the same limitations as the apparatus of claim 14. Therefore, claim 29 is rejected under the same rationale.

Claims 31-33 recite a method with substantially the same limitations as the utility of claims 22-24, respectively. Therefore, the claims are rejected under the same rationale.

Regarding claim 38, Bach '739/'660/Francis teaches deploying the at least one XMI file and middleware application to servers configured to enable transactional communication between the client application and the MFS-based IMS application (see Bach '739 paragraph [0014]; "*Places the CGI-BIN executable file in the directory which was user specified in IMS Web Studio (`Web Server's CGI Path`). This directory must be the same directory that their Web server uses to look for CGI-BIN scripts*", the CGI-BIN directory is the repository that stores the generated middleware applications to interface with the MFS-based IMS applications).

Claim 39 recites an apparatus with substantially the same limitations as the utility of claim 20. Therefore, claim 39 is rejected under the same rationale.

Claim 40 recites an article of manufacture with substantially the same limitations as the utility of claim 20. Therefore, claim 40 is rejected under the same rationale.

Claims 7 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bach '739 (5,781,739) *supra*, Bach '660 (6,141,660) *supra*, and Narayan, United States Patent Application Publication number 2002/0078255.

Regarding claim 7, Bach '739/'660 teach all the limitations of claim 7 except that the command-line interface comprises a plurality of modes, each mode comprising a different level of user interaction. However, it was well known in the art at the time the invention was made that some command-line interfaces comprised a plurality of modes, each mode comprising a different level of user interaction. Narayan teaches an command line interface with multiple modes of operation (see Narayan paragraph [0044]; "*Besides enabling the interaction with the software in real time, the command line interface also facilitated batch mode usage of software*"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to provide command line interface modes as taught by Narayan in the code generating interface as taught by Bach '739/'660 in order to provide another method of interaction with the invention.

Regarding claim 8, Bach '739/'660/Narayan teaches that one mode comprises a batch mode that reads the parameter set from persistent storage (see Narayan paragraph [0044]; "*Besides enabling the interaction with the software in real time, the command line interface also facilitated batch mode usage of software*").

Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bach '739 (5,781,739) *supra*, Bach '660 (6,141,660) *supra*, and Dan et al. (hereinafter Dan), United States Patent number 6,560,639.

Regarding claim 9, Bach '739/'660 teaches all the limitations of claim 9 except an error module configured to present an error message in response to an error condition triggered by the import module, the metadata generator, or the code generator. However, including an error module to respond to application errors was well known in the art at the time the invention was made, as evidenced by Dan (see column 5 lines 1-11; "*error report manager may report any error in intended user changes to a requested web page*"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include an error module as described by Dan in the invention of Bach '739/'660 for the purpose of handling error conditions in the application.

Claims 10 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bach '739 (5,781,739) *supra*, Bach '660 (6,141,660) *supra*, and Snover et al. (hereinafter Snover), United States Patent Application Publication number 2004/0230987.

Regarding claim 10, Bach '739/'660 teaches all the limitations of claim 10 except that the import module is configured to import a plurality of MFS-based IMS source files in response to a single parameter. However, it was well known in the art at the time the invention was made that a wildcard character can be entered on many command line interfaces to specify several files using a single parameter, as indicated by Snover (see Snover paragraph [0054]; "*the service could perform wildcard expansion on a filename entered as "A*" on the command line. Before the second pass, the Name member contains "A*" as it was entered on the command line. During the second pass, the Filename service may locate a set of files starting with A and store them in the Arraylist F*"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the wildcard expansion technique of Snover with the web interface generator of Bach '739/'660 in order to make importing several files at the command line less tedious.

Claims 16-18, 25, 26, 34, and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bach '739 (5,781,739) *supra*, Bach '660 (6,141,660) *supra*, Francis (6,665,861) *supra*, and Narayan, United States Patent Application Publication number 2002/0078255.

Regarding claim 16, Bach '739/'660/Francis teaches every limitation of claim 16 except that the command-line interface comprises a plurality of modes, each mode involving a different level of user interaction (see Narayan paragraph [0044]; "*Besides enabling the interaction with the software in real time, the command line interface also facilitated batch mode usage of software*"). However, it was well known in the art at the time the invention was made that some command-line interfaces comprised a plurality of modes, each mode comprising a different level of user interaction. Narayan teaches an command line interface with multiple modes of operation (see Narayan paragraph

[0044]; "*Besides enabling the interaction with the software in real time, the command line interface also facilitated batch mode usage of software*"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to provide command line interface modes as taught by Narayan in the code generating interface as taught by Bach '739/'660/Francis in order to provide another method of interaction with the invention.

Regarding claim 17, Bach '739/'660/Francis/Narayan teaches that one mode prompts a user for each parameter of the parameter set (see Narayan paragraph [0326]; "*presents the user with a user interface that prompts for user information when appropriate server method is invoked*").

Regarding claim 18, Bach '739/'660/Francis/Narayan teaches that the command-line interface is configured to be executed by a separate software module (see Narayan paragraph [0058]; "*The UI client has a user interface component that is either graphical or command line*").

Claim 25 recites a utility with substantially the same limitations as the apparatus of claim 16. Therefore, claim 25 is rejected under the same rationale.

Regarding claim 26, Bach '739/'660/Francis/Narayan teaches that one mode comprises a batch mode that reads the parameter set from persistent storage (see Narayan paragraph [0044]; "*Besides enabling the interaction with the software in real time, the command line interface also facilitated batch mode usage of software*").

Claim 34 recites a method with substantially the same limitations as the apparatus of claim 16. Therefore, claim 34 is rejected under the same rationale.

Claim 35 recites a method with substantially the same limitations as the utility of claim 26. Therefore, claim 35 is rejected under the same rationale.

Claims 27 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bach '739 (5,781,739) *supra*, Bach '660 (6,141,660) *supra*, Francis (6,665,861) *supra*, and Dan et al. (hereinafter Dan), United States Patent number 6,560,639.

Regarding claim 27, Bach '739/'660/Francis teaches all the limitations of claim 9 except an error module configured to present an error message in response to an error condition triggered by the import module, the metadata generator, or the code generator. However, including an error module to respond to application errors was well known in the art at the time the invention was made, as evidenced by Dan (see Dan column 5 lines 1-11; "*error report manager may report any error in intended user changes to a requested web page*"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include an error module as described by Dan in the invention of Bach '739/'660/Francis for the purpose of handling error conditions in the application.

Claim 36 recites a method with substantially the same limitations as the utility of claim 27. Therefore, claim 36 is rejected under the same rationale.

Claims 28 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bach '739 (5,781,739) *supra*, Bach '660 (6,141,660) *supra*, Francis (6,665,861) *supra*, and Snover et al. (hereinafter Snover), United States Patent Application Publication number 2004/0230987.

Regarding claim 28, Bach '739/'660/Francis teaches all the limitations of claim 10 except that the import module is configured to import a plurality of MFS-based IMS source files in response to a single parameter. However, it was well known in the art at the time the invention was made that a wildcard character can be entered on many command line interfaces to specify several files using a single parameter, as indicated by Snover (see Snover paragraph [0054]; "*the service could perform wildcard expansion on a filename entered as "A*" on the command line. Before the second pass, the Name member contains "A*" as it was entered on the command line. During the second pass, the Filename service may locate a set of files starting with A and store them in the Arraylist F*"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the wildcard expansion technique of Snover with the web interface generator of Bach '739/'660/Francis in order to make importing several files at the command line less tedious.

Claim 37 recites a method with substantially the same limitations as the utility of claim 28. Therefore, claim 37 is rejected under the same rationale.

Response to Arguments

Applicants note that Chiang (2004/0054969) is invalid prior art for a 35 USC § 103(a) rejection. The examiner agrees. Accordingly, the Chiang reference has been withdrawn from use as a 35 USC § 103(a) reference.

Applicants assert that Bach '739 (5,781,739) does not teach an interface which receives a parameter set as a single input and executes the listed modules. The examiner agrees. However, Applicant's arguments with respect to claims 1, 2, 4-20, 22-29, and 31-40 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Stephen Alvesteffer whose telephone number is (571) 270-1295. The examiner can normally be reached on Monday-Friday 9:30AM-6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong can be reached on (571)272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Stephen Alvesteffer
Examiner
Art Unit 2173

1-5-2008
TADESSE HAILU
PRIMARY EXAMINER